VIERRA MAGEN
MARCUS & DENIRO LLP
INTELLECTUAL PROPERTY LAW

575 MARKET ST. • SUITE 2500
SAN FRANCISCO, CA 94105-2871
TELEPHONE 415.369.9660
FACSIMILE 415.369.9665
WWW.VIERRAMAGEN.COM

FACSIMILE TRANSMITTAL

TO: Examiner Zheng Wei	FROM: Michelle Esteban
COMPANY: USPTO	DATE: September 25, 2007
FAX NUMBER: 571-270-2059	TOTAL NO. OF PAGES, INCLUDING COVER: 9
PHONE NUMBER: 571-270-1059	SENDER'S REFERENCE NUMBER:
RE: Application No. 10/642,360	YOUR REFERENCE NUMBER:

IF YOU DO NOT RECEIVE ALL PAGES, PLEASE CALL: MICHELLE ESTEBAN AT 415.369.9660.

NOTES/COMMENTS:

Mr. Wei:

Please see attached claim amendments.

Thank you so much for your patience in this matter. Please feel free to contact me if you have any questions or concerns.

Thank you,

[illegible]

Michelle Esteban
Reg. No. 59,880

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (currently amended) A method to provide for evaluating code expressions, comprising:

receiving code for a program, said code includes one or more expressions and one or more markers that specify one or more times when said one or more expressions should be evaluated during execution of said program; and

automatically providing additional functionality to said code for said program, said additional functionality evaluates said one or more expressions during execution of said program at said one or more times specified by said one or more markers.

2. (original) A method according to claim 1, wherein:

said one or more markers specify when said one or more expressions should be evaluated during execution of said program independent from a context of where said expressions are used.

3. (currently amended) A method according to claim 1, wherein:

said markers ~~can~~ that specify one or more times further indicate that a particular expression should be evaluated immediately, once or always.

4. (previously presented) A method according to claim 3, wherein:

failure of a marker to indicate that a particular expression should be evaluated immediately or once defaults to an indication indicating that said particular expression should be evaluated always.

5. (previously presented) A method according to claim 1, wherein:

said one or more expressions are constraints for variables; and

said step of automatically providing additional functionality to said code includes adding code that creates an object for each constraint, adds functions to said object that set said variables, and adds functions that set dependencies for said expressions.

6. (original) A method according to claim 1, wherein:

said code for said program is XML code.

7. (original) A method according to claim 1, wherein:

said step of automatically providing additional functionality to said code includes compiling said code.

8. (original) A method according to claim 1, further comprising:

receiving a request for content via a network;

transmitting said code with said additional functionality to a client via said network; and

executing said code with said additional functionality at said client.

9. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method to provide for evaluating expressions, said method comprising:

accessing code for a program, said code includes one or more expressions and one or more markers that specify <u>one or more times</u> when said one or more expressions should be evaluated during execution of said program; and

automatically providing for an evaluation of said one or more expressions during execution of said program at <u>said</u> one or more times specified by said one or more markers.

10. (original) One or more processor readable storage devices according to claim 9, wherein:

said one or more markers specify when said one or more expressions should be evaluated during execution of said program independent from a context of where said expressions are used.

11. (currently amended) One or more processor readable storage devices according to claim 9, wherein:

said markers ~~can~~ that specify one or more times further indicate that a particular expression should be evaluated immediately, once or always.

12. (original) One or more processor readable storage devices according to claim 9, wherein:

said step of automatically providing for an evaluation includes compiling said code.

13. (currently amended) A method to provide for evaluating code expressions, comprising:

receiving code for a program, said code includes one or more expressions and one or more markers that specify one or more times when said one or more expressions should be evaluated during execution of said program; and

evaluating said one or more expressions during execution of said program at said one or more times specified by said one or more markers.

14. (original) A method according to claim 13, wherein:

said one or more markers specify when said one or more expressions should be evaluated during execution of said program independent from a context of where said expressions are used.

15. (currently amended) A method according to claim 13, wherein:

said markers ~~can~~ that specify one or more times further indicate that a particular expression should be evaluated immediately, once or always.

16. (original) A method according to claim 13, wherein:

said code for said program is XML source code.

17. (original) A method according to claim 13, wherein:
said code for said program is object code.

18. (currently amended) A method to provide for evaluating <u>code</u> expressions, comprising:

accessing code that includes an expression defining a first variable <u>and an indicator for said expression</u>, said expression is dependent on a changeable item <u>within said expression, said indicator specifies one or more times when said expression should be evaluated</u>; and

compiling said code, said step of compiling said code adds additional functionality to said code, said additional functionality evaluates said expression when said <u>changeable</u> item <u>within said expression</u> changes <u>during execution based on said indicator</u> and updates said first variable.

19. (original) A method according to claim 18, wherein:

said expression is part of a constraint for said first variable;

said step of compiling includes creating an object for said constraint, adding a first function to said object that sets said first variable, determining dependency of said expression and adding a second function for said dependency.

20. (original) A method according to claim 19, wherein:

said additional functionality includes code that adds said first function to an object for said first variable and code that provides a pointer to said first function to an object for said changeable item to be called by said object for said changeable item when said changeable item changes.

21. (cancelled)

22. (currently amended) One or more processor readable storage devices having

processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method to provide for evaluation of expressions, said processor readable code comprising:

preexisting functionality that evaluates expressions when a dependency changes and updates a variable based on said expression;

code that accesses first code, said first code includes a first expression defining a first variable and an indicator for said expression, said first expression is dependent on a first dependency within said first expression, said indicator specifies one or more times said expression should be evaluated when said first dependency changes; and

code that combines said preexisting functionality with said first code so that when said first code is executed said first variable is updated by said first expression when said first dependency within said first expression changes.

23. (original) A method according to claim 22, wherein:

said first expression is part of a constraint for said first variable; and

said code that combines said preexisting functionality with said first code creates an object for said constraint, adds a first function to said object that sets said first variable, determines dependency of said first expression and adds a second function for said dependency to said object.

24. (original) A method according to claim 22, wherein:

said preexisting functionality includes code that adds said first function to an object for said first variable and code that provides a pointer to said first function to an object for said first dependency to be called by said object for said first dependency when said first dependency changes.

25. (cancelled)

26. (currently amended) A method to provide for evaluating code expressions,

comprising:

receiving code that includes an expression defining a first variable and an indicator for said expression, said expression is dependent on a changeable item within said expression, said indicator specifies one or more times when said expression should be evaluated; and

automatically providing additional functionality to said code, said additional functionality evaluates said expression when said changeable item within said expression changes during execution based on said indicator and updates said first variable.

27. (original) A method according to claim 26, wherein:

said expression is part of a constraint for said first variable; and

said step of automatically providing includes creating an object for said constraint, adding a first function to said object that sets said first variable, determining dependency of said expression and adding a second function for said dependency to said object.

28. (original) A method according to claim 27, wherein:

said additional functionality includes code that adds said first function to an object for said first variable and code that provides a pointer to said first function to an object for said changeable item to be called by said object for said changeable item when said changeable item changes.

29. (cancelled)

30. (original) A method according to claim 26, further comprising:

requesting said code by an Internet client;

transmitting said code with said additional functionality to said Internet client after said step of automatically providing; and

executing said code with said additional functionality using said Internet client.

31. (currently amended) One or more processor readable storage devices having

processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method to provide for evaluating expressions, said method comprising:

accessing code that includes an expression defining a first variable and an indicator for said expression, said expression is dependent on a changeable item within said expression, said indicator specifies one or more times when said expression should be evaluated; and

automatically providing preexisting additional functionality to said code, said preexisting additional functionality evaluates said expression when said changeable item within said expression changes during execution based on said indicator and updates said first variable.

32. (original) One or more processor readable storage devices according to claim 31, wherein:

said expression is part of a constraint for said first variable; and

said step of automatically providing includes creating an object for said constraint, adding a first function to said object that sets said first variable, determining dependency of said expression and adding a second function for said dependency to said object.

33. (original) One or more processor readable storage devices according to claim 32, wherein:

said additional functionality includes code that adds said first function to an object for said first variable and code that provides a pointer to said first function to an object for said changeable item to be called by said object for said changeable item when said changeable item changes.

34. (cancelled)

35. (original) One or more processor readable storage devices according to claim 31, wherein:

said preexisting additional functionality prevents circular evaluation.

36. (currently amended) An apparatus that provides for evaluation of code expressions, comprising:

a processor readable storage device; and

one or more processors in communication with said processor readable storage device, said one or more processors perform a method comprising the steps of:

accessing code that includes an expression defining a first variable and an indicator for said expression, said expression is dependent on a changeable item within said expression, said indicator specifies one or more times when said expression should be evaluated, and

automatically providing preexisting additional functionality to said code, said preexisting additional functionality evaluates said expression when said changeable item within said expression changes during execution based on said indicator and updates said first variable.

37. (original) An apparatus according to claim 36, wherein:

said expression is part of a constraint for said first variable;

said step of automatically providing includes creating an object for said constraint, adding a first function to said object that sets said first variable, determining dependency of said expression and adding a second function for said dependency to said object; and

said additional functionality includes code that adds said first function to an object for said first variable and code that provides a pointer to said first function to an object for said changeable item to be called by said object for said changeable item when said changeable item changes.